IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 10/081,755
Filing Date 02/19/02
Inventorship ..... Dharmarajan
Applicant ..... Microsoft Corporation
Group Art Unit 2134
Examiner ..... Tran, Tongoc
Attorney's Docket No. MS1-1055US
Title: Request Persistence During Session Authentication

RESPONSE TO OFFICE ACTION DATED 10/03/2005

To: Commissioner of Patents
Alexandria, VA 22313-1450

From: Kayla D. Brant (Tel. 509-324-9256; Fax 509-323-8979)
Customer No. 22801

AMENDMENTS TO THE CLAIMS

No claims are canceled.

No claims are added.

Claims 1, 12-14, 23, and 25-27 are amended.

Claims 1-29 are pending.

1. (Currently amended) A method comprising:

establishing an authenticated session with a client;

receiving a request from the client;

determining whether the session is still authenticated;

in an event that the session is ~~not~~ no longer authenticated, persisting the request from the client as a pending request; and

in an event that the session is subsequently re-authenticated, processing the pending request.

2. (Original) The method of claim 1 wherein the determining comprises verifying an authentication token associated with the client.

3. (Original) The method of claim 2 wherein the verifying comprises verifying that the authentication token has not timed out.

4. (Original) The method of claim 2 wherein the authentication token is a cookie stored by the client.

5. (Original) The method of claim 2 wherein the authentication token is part of the request received from the client.

6. (Original) The method of claim 2 wherein the authentication token is encrypted.

7. (Original) The method of claim 1 wherein persisting the request comprises storing the request in a file.

8. (Original) The method of claim 1 wherein persisting the request comprises storing the request in a database.

9. (Original) The method of claim 1 further comprising, after persisting the request, directing the client to authenticate the session.

10. (Original) The method of claim 9 wherein directing the client to authenticate the session comprises:

directing the client to a login module; and

directing the client to an address associated with the pending request.

11. (Original) The method of claim 10 wherein the address associated with the pending request is a URL.

12. (Currently amended) A method comprising:

establishing an authenticated session with a server;

submitting a request to the server;

receiving an indication that the session is ~~not~~ no longer authenticated;

obtaining a session re-authentication; and

receiving an indication that the request has been processed.

13. (Currently amended) A system comprising:

an authentication verifier configured to determine whether an initially authorized session associated with a client is still authorized;

a client interface configured to receive a request from the client;

a pending request store configured to maintain the request in an event that the session is not authorized; and

a processing unit configured to process the request that is maintained in an event that the session is re-authorized.

14. (Currently amended) The system of claim 13 further comprising an authentication redirect generator configured to generate an instruction to redirect the client to obtain re-authorization for the session.

15. (Original) The system of claim 14 wherein the instruction is a URL.

16. (Original) The system of claim 14 wherein the authorization is an authentication token.

17. (Original) An application server comprising the system as recited in claim 13.

18. (Original) A system comprising:

a client interface configured to receive a request from a client;

an authentication token verifier configured to determine whether an authentication token associated with the client is valid;

a pending request store configured to store the request in an event that the authentication token associated with the client is not valid; and

an authentication redirect generator configured to generate an instruction to redirect the client to obtain a valid authentication token.

19. (Original) The system of claim 18 wherein the authentication token verifier is further configured to determine whether the authentication token has expired.

20. (Original) The system of claim 18 wherein the authentication redirect generator is further configured to direct the client to access the request that is stored.

21. (Original) The system of claim 18 wherein the pending request store is a database.

22. (Original) A system comprising:

means for receiving a request from a client;

means for determining whether an authentication token associated with the client is valid;

means for storing the request in an event that the authentication token is not valid; and

means for generating an instruction to redirect the client to obtain a valid authentication token.

23. (Currently amended) A system comprising:

a client;

an application server configured to:

establish a session with an authenticated client;

receive a request from the client;

maintain the request as a pending request in an event that the client is ~~not~~ no longer authenticated; and

direct the client to re-obtain authentication;

the client being configured to obtain authentication from an authentication entity in response to direction from the application server, and the client further configured to subsequently access the pending request; and

upon client access to the pending request, the application server being further configured to process the pending request.

24. (Original) The system of claim 23 wherein the application server and the authentication entity are implemented as one server.

25. (Currently amended) One or more computer-readable media comprising computer executable instructions that, when executed, direct a computing system to:

establish a session with an authenticated client;

receive a request from ~~a~~ the client;

determine whether the client is still authenticated;

in an event that the client is ~~not~~ no longer authenticated, persist the request; and

in an event that the client is subsequently re-authenticated, process the request that is persisted.

26. (Currently amended) The one or more computer-readable media of claim 25 further comprising computer executable instructions that, when executed, direct a computing system to:

in the event that the client is ~~not~~ no longer authenticated,

redirect the client to re-obtain authentication; and

direct the client to the request that is persisted.

27. (Currently amended) One or more computer-readable media comprising computer executable instructions that, when executed, direct a computing system to:

establish a session with an authenticated client;

receive a request from ~~a~~ the client;

determine whether an authentication token associated with the client is still valid;

store the request if the authentication token is ~~not~~ no longer valid; and

generate an instruction to redirect the client.

28. (Original) The one or more computer-readable media of claim 27 wherein the instruction comprises an instruction to redirect the client to obtain a valid authentication token.

29. (Original) The one or more computer-readable media of claim 28 wherein the instruction further comprises an instruction to redirect the client to the request that is stored upon the client obtaining the valid authentication token.

REMARKS

In view of the following remarks, Applicant respectfully requests reconsideration and allowance of the subject application. No claims have been added or cancelled. Claims 1, 12-14, 23, and 25-27 have been amended. Claims 1-22 are pending.

Rejections to the Claims

35 U.S.C. 103

Claims 1-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over United States Patent No. 6,678,733 to Brown et al. (herein referred to as "Brown") in view of United States Patent Application Publication No. 2002/0112083 to Joshi et al. (herein referred to as "Joshi").

Claims 1, 12-14, 23, and 25-27 have been amended to more clearly describe the claimed subject matte, support for which may be found throughout the specification and drawings as filed. For example, Applicant's application describes a re-authentication system implemented on a server computer system that provides increased security without disrupting user workflow in a client-server environment. When a request is submitted from the client to the server, the re-authentication system verifies that the session is secure. If the re-authentication system cannot verify that the session is secure, the system persists (e.g., saves or maintains) the request and directs the client to re-authenticate the session. When the client session is re-established, the re-

authentication system directs the server to process the saved request, instead of requiring that the request be re-submitted from the client. (*Application*, page 2, line 25 – page 3, line 8.) Specifically, claim 1 recites a method comprising:

establishing an authenticated session with a client;

receiving a request from the client;

determining whether the session is still authenticated;

in an event that the session is no longer authenticated, persisting the request from the client as a pending request; and

in an event that the session is subsequently re-authenticated, processing the pending request.

Brown and Joshi both describe establishing an authenticated session with a client, however, the combination of Brown and Joshi does not teach or suggest, "determining whether the session is ***still*** authenticated; in an event that the session is no longer authenticated, persisting the request from the client as a pending request; and in an event that the session is subsequently re-authentication, processing the pending request," as recited in claim 1.

Brown describes a centralized authentication and authorization system for providing users access to groups of network resources. Specifically, Brown describes a system and method for authenticating and authorizing users seeking access to resources on a network. (*Brown*, col. 1, lines 22-24.) Brown describes a walled garden via which multiple network-based services may be available. (*Brown*, col. 2, lines 46-50.) When a user wishes to access a

service in the walled garden, the client sends an HTTP request including a ticket granting access to the walled garden. If the client does not provide a ticket or the ticket is invalid, the HTTP request is denied. (*Brown*, col. 2, line 66-col. 3, line 6.) Brown describes establishing an authenticated session with a client, but Brown does not teach or suggest "receiving a request from the client; determining whether the session is still authenticated; in an event that the session is no longer authenticated, persisting the request from the client as a pending request; and in an event that the session is subsequently re-authenticated, processing the pending request," as recited in claim 1. The Office agrees that "Brown does not disclose the server storing the request and processing the request after the client is authenticated." (*Office Action*, page 2-3.)

Joshi describes a system that provides centralized authentication, authorization, and auditing services for resources hosted on or available to one or more web servers. (*Joshi*, paragraph [0095].) An end user enters a URL of a requested resource residing in a protected policy domain. The request is intercepted, and if the end user has not already been authenticated, then a challenge is issued to the browser for log-on information. If the log-on information satisfies the authentication criteria, and the user is also authorized to access the requested resource, then access to the resource is granted. (*Joshi*, paragraph [0100].) When a user first requests a protected resource, the user is challenged according to the authentication scheme. If the user satisfies the authentication rule, an encrypted authentication cookie is passed to the

user's browser indicating a successful authentication. (*Joshi*, paragraph [0155].) Once authenticated, a user can explicitly log out, causing authentication cookies cached (or otherwise stored) by the user's browser to be destroyed or become invalid. Authentication cookies can also be set by an administrator to be destroyed after a maximum idle time has elapsed between requests to protected resources. (*Joshi*, paragraph [0156].)

Joshi specifically indicates that authentication cookies may be destroyed after a maximum ***idle*** time has elapsed. However, Applicant's application points out that this presents a potential security problem that can be overcome by the system described in Applicant's application. (Application, page 1, lines 15-24.) To overcome this security problem, an authentication cookie can be set to expire after a particular time period (idle or not), resulting in a session that is no longer authenticated. However, this may result in a user workflow disruption. (Application, page 1, lines 6-14.) The workflow disruption is overcome, as indicated in claim 1, by "in an event that the session is no longer authenticated, persisting the request from the client as a pending request; and in an event that the session is subsequently re-authenticated, processing the pending request."

The Office claims that "Joshi discloses that when the web server received the client's http request, it stores the request in the server; Joshi discloses the process of performing authentication actions and administrator can set up a redirect URL for authentication success/failure events." (*Office Action*, page 3.) The Office cites Joshi, paragraphs [0241 – 0242].

The request that is stored according to Joshi is a request to access a protected resource. The request is intercepted by a centralized authentication service, which stores the request while authentication is performed. If the user is successfully authenticated, the saved request is then forwarded to its intended recipient, thus establishing a secure session between the protected resource and the client.

The Office also takes Official Notice that "allowing a user to request a service and prompting user to be authenticated before granting user the request is old and well known." While prompting a user to be authenticated before granting the user access to a service may be well known, Applicant is not aware of any systems in which a secure session can be established with a client, and then later, a request is received from the client, it is determined that the session is no longer authenticated (e.g., an authentication token has timed out), and the request is maintained to be processed if the client successfully re-authenticates the session.

There is no suggestion in Joshi or Brown that once an authenticated session is established with a client, that the steps of "receiving a request from the client; determining whether the session is still authenticated; in an event that the session is no longer authenticated, persisting the request from the client as a pending request; and in an event that the session is subsequently authenticated, processing the pending request," be performed, as recited in claim 1. Accordingly, claim 1 is allowable over the combination of Brown and Joshi.

Claims 2-11 are allowable by virtue of their dependency on claim 1.

Claims 12, 13, 18, 22, 23, 25, and 27 are allowable for reasons similar to those stated above with reference to claim1.

Claims 14-17, 19-21, 24, 26, 28, and 29 are allowable by virtue of their respective, direct or indirect, dependency on claims 13, 18, 22, 23, 25, and 27.

Conclusion

Claims 1-29 are believed to be in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the present application. Should any issue remain that prevents immediate issuance of the application, the Examiner is encouraged to contact the undersigned agent to discuss the unresolved issue.

Respectfully Submitted,
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

Dated: 12/15/05

Name: Kayla D. Brant
Reg. No. 46,576
Phone No. (509) 324-9256 ext. 242